etruecolor

February 23, 2011

Abstract

Generate three-color coded spatial image from scalar event table attribute

1 Instruments/Modes

Instrument	Mode

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

etruecolor generates from a set of input event list(mos and pn) a spatial image in which the value of a specified scalar event attribute (e.g. energy) is three-color coded. The coloring process is driven by three fundamental color curves (red, green, blue) that are constructed from the contents of a pre-defined input color table (RGB_Scheme).

The energy bands (coded in the pre-defined color table) for the red, green and blue channels of the true color image have been chosen so that a power law spectrum with photon spectral index 1.7, absorbed by galactic gas with neutral hydrogen column density of 10^{20} cm^{-2}, results in approximately the same number of counts in each band.

Channel	energy Range
Red	300-700 eV
Green	700-1200 eV
blue	1200-7000 eV

Please note: The actual generation of the red, green, and blue component images is done through the task **evselect**. **etruecolor** therefore inherits some of **evselect**'s image extraction parameters which allows to control the image generation process, e.g., binning, windowing, etc.

The default value for the image scale creation is **4 arcsec/pixel**, that is equivalent to a`ximagebinsize` equals to 80.

3.1 Filtering and vigneting correction

etruecolor task filters the input tables using the `min` and `max` parameter.

In order to flat field the images, **etruecolor** runs **evigweight** task. This task creates a WEIGHT column that it is used for vigneting corretion.

3.2 Brightness limits and scaling

For a pipeline production of true color images it seems to be more adequate to use the same scaling and cuts in all images, instead of trying to optimize for every particular case. This approach may result in color saturation for very bright sources and/or missing the weakest sources. Also it permits a direct comparison between different images, independently on their effective exposure time.

Currently proposed limits correspond to 2.25 and 675 counts/ks/arcmin^2 and the scaling is logarithmic. This limits correspond to 0.01 and 3 counts/ks/pixel (assuming an image scale of 4 arcsec/pixel).

4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints
tablelist	yes	table		name of existing table

List of tables that contains the event data - need to at least contain the columns named with the `xcolumn`, `ycolumn`, `ecolumn` parameters.

colortable	no	string	RGB_Scheme	name of predefined color table

The energy bands for the pre-defined color table are explained in the text.

min	no	real	300	

Explicit minimum value of `ecolumn` column. This value is used for filtering.

max	no	real	7000	

Explicit maximum value of `ecolumn` column. This value is used for filtering.

ximagebinsize	no	real	80	> 0

passed to **evselect** as parameter `ximagebinsize`

withxranges	no	boolean	false	true|false

passed to **evselect** as parameter `withxranges`

ximagemin	no	real	1	> 0

passed to **evselect** as parameter `ximagemin`

ximagemax	no	real	600	> 0

passed to **evselect** as parameter `ximagemax`

yimagebinsize	no	real	80	> 0

passed to **evselect** as parameter `yimagebinsize`

withyranges	no	boolean	false	true|false

passed to **evselect** as parameter `withyranges`

yimagemin	no	real	1	> 0

passed to **evselect** as parameter `yimagemin`

yimagemax	no	real	600	> 0

passed to **evselect** as parameter `yimagemax`

scalmin	no	real	2.25	

Min value for log scaling (counts/ks/arcmin^2).

scalmax	no	real	675	

Max value for log scaling (counts/ks/arcmin^2)

fileset	no	string	default	name of output data set

The name of the data set the color images as count rates. This file is a RGB cube ready to be displayed with Ds9. Leaving the default value, **etruecolor** task creates the following file name: PppppppooeeEPX000RGBIMA0000.FIT, where pppppppooee is the extended observation Id.

outputchoice	no	string	ppmfile	dataset|ppmfile

If set to *dataset* image is written to a data set whose name is given via parameter `colorset`. Otherwise, image is written in PPM format to standard file named via `ppmfile`.

colorset	no	string	default	name of data set

The name of the data set the color image shall be written to if `outputchoice`=*dataset*. Depending on the value of `ascube` the data will either be written to three separate arrays corresponding to the red, green, and blue components or three slices of a 3-D data cube in the primary array, respectively. The data set can be read and the contents displayed with Ds9. Leaving the default value, **etruecolor** task creates the following file name: PppppppooeeEPX000RGBCOL0000.FIT, where pppppppooee is the extended observation Id.

ppmfile	no	string	default	name of file

If `outputchoice`=*ppmfile* the name of a PPM data file that the color image shall be written to. If `ppmfile`=*stdout* the data shall be written to standard output. Leaving the default value, **etruecolor** task creates the following file name: PppppppooeeEPX000RGBIMA0000.PPM, where pppppppooee is the extended observation Id.

ascube	no	boolean	false	false|true

Boolean parameter determining whether the red, green, and blue component images are to be written as three separate array extensions to the data set designated with `colorset` or as three slices of a single 3-dimensional data cube in the primary array.

5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

MinMaxEqual *(error)*
The minimum and maximum values of the scalar attribute data are equal. This must not happen because in this case the mapping onto the color table *level* range $[0, 1]$ is undefined.

In addition all error and warning messages of the task **evselect** and package **dal** can occur.

6 Input Files

1. event file containing x and y position and scalar attribute of a set of events.
2. color set with a table that gives the intensities of red, green and blue as a function of a scalar value.

7 Output Files

1. PPM (portable pixmap) file or
2. data set containing either
   - three arrays with R/G/B components respectively
   - R/G/B images as slices in primary array

The data set format is readable by the image viewer Ds9 in version 2.3 or later. Generated PPM pixel maps can be visualized with the display program `xv`.

8 Algorithm

```
read RGB color curves
setup temporary table with columns red/green/blue
foreach event
  red = linearInterpolate(redcurve,energy)
  green = linearInterpolate(greencurve,energy)
  blue = linearInterpolate(bluecurve,energy)
foreach {red, green, blue}
  construct component image with evselect and divide by livetime
combine partial images
if (log)
  foreach pixel
    r,g,b = max ( log(r,g,b) - log(maxValue) + decades, 0)
normalize to 255
if (withcolorset)
   write image to data set
else
   write image in PPM format
```

9 Comments

References